IN THE CLAIMS

The following listing of claims replaces all prior claim versions and listings.

1. (Currently Amended) A hardware description language based circuit design verification ~~verifying~~ system comprising:

a storage unit configured to ~~which~~ stores a source program for hardware description in a programming~~program~~ language;

an output unit; and

a processor configured to~~which~~ detect a portion of said source program whose compiled code obtained using a software compiler is different in logic ~~interpretation between a case of compiling said source program using a compiler and a case of~~ from the portion's behavioral synthesis-produced module, and to outputs existence of said source program portion to said output unit.

2. (Currently Amended) The ~~hardware description verifying~~ system according to claim 1, wherein said processor detects said source program portion in which a variable of a register type is referred to at a clock timing event following ~~after~~ substitution to said variable ~~at said clock~~ timing event.

3. (Currently Amended) The ~~hardware description verifying~~ system according to claim 2, wherein said processor comprises:

a data signal list storage section; and

a processing section which sequentially reads out statements ~~sentences~~ of said source program, deletes a storage content in said data signal list storage section when the read out statement ~~sentence~~ indicates a clock boundary, stores said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event, and outputs the existence of said read out ~~sentence~~statement to said output unit when said read out ~~sentence~~statement refers to said variable which is stored in said data signal list storage section.

4. (Currently Amended) The ~~hardware description verifying~~ system according to claim 1, wherein said processor detects said source program portion in which substitution to a variable of a non-overwrite type is carried out twice or more at a clock timing event.

5. (Currently Amended) The ~~hardware description verifying~~ system according to claim 4, wherein said processor comprises:

a data signal list storage section; and

a processing section which sequentially reads out ~~sentences~~statements of said source program, deletes a storage content in said signal list storage section when the read out ~~sentence~~statement indicates a clock boundary, stores said variable in said signal data list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event and said variable is not stored in said data signal list storage section, and outputs the existence of said read out ~~sentence~~statement to said output unit when said read out ~~sentence~~statement indicates the substitution to said

variable at said clock timing event and said variable is stored in said data signal list storage section.

6. (Currently Amended) The ~~hardware description verifying~~ system according to claim 1, wherein said processor detects said source program portion in which a variable for a non-register type is referred to at a clock timing event without substitution of said value to said variable at said clock timing event.

7. (Currently Amended) The ~~hardware description verifying~~ system according to claim 6, wherein said processor comprises:

a data signal list storage section; and

a processing section which sequentially reads out ~~sentence~~statements of said source program, deletes a storage content in said data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary, stores said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event, and outputs the existence of said read out ~~sentence~~statement to said output unit when said read out ~~sentence~~statement refers to said variable which is not stored in said data signal list storage section, at said clock timing event.

8. (Currently Amended) The ~~hardware description verifying~~ system according to claim 1, wherein said processor detects said source program portion in

which a variable of a wiring line is referred to at a clock timing event and then a value is substituted to said variable at said clock timing event.

9. (Currently Amended) The ~~hardware description verifying~~ system according to claim 8. wherein said processor comprises:

a data signal list storage section; and

a processing section which sequentially reads out ~~sentence~~statements of said source program, deletes a storage content in said data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary, stores said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event, and outputs the existence of said read out ~~sentence~~statement to said output unit when said read out ~~sentence~~statement refers to said variable which is not stored in said data signal list storage section, at said clock timing event.

10. (Currently Amended) The ~~hardware description verifying~~ system according to claim 1, wherein said processor detects said source program portion in which a logical operator is used and a right operand of the operator type includes a variable with substitution.

11. (Currently Amended) A hardware description language based circuit design verification ~~verifying~~ method comprising:

(a) detecting a portion of a source program whose compiled code based on a compiler output is different in logic ~~interpretation between a case of compiling said source program using a compiler and a case of~~ from the portion's behavioral synthesis-produced module, said source program for hardware description being described in a program language; and

(b) ~~alarming~~ signaling an existence of said source program portion.

12. (Currently Amended) The ~~hardware description verifying~~ method according to claim 11, wherein said source program portion is a portion in which a variable of a register type is referred to a clock timing event after substitution to said variable at said clock timing event.

13. (Currently Amended) The ~~hardware description verifying~~ method according to claim 12, wherein said (a) detecting comprises:

sequentially reading out ~~sentence~~statements of said source program;

deleting a storage content in a data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event; and

detecting said read out ~~sentence~~statement as said source program portion when said read out ~~sentence~~statement refers to said variable which is stored in said data signal list storage section.

14. (Currently Amended) The ~~hardware description verifying~~ method according to claim 11, wherein said source program portion is a portion in which substitution to a variable of a non-overwrite type is carried out twice or more at a clock timing event.

15. (Currently Amended) The ~~hardware description verifying~~ method according to claim 14, wherein said (a) detecting comprises:

sequentially reading out ~~sentence~~statements of said source program;

deleting a storage content in a data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event and said variable is not stored in said data signal list storage section; and

detecting said source program portion when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event and said variable is stored in said data signal list storage section.

16. (Currently Amended) The ~~hardware description verifying~~ method according to claim 11, wherein said source program portion is a portion in which a variable for a non-register type is referred to at a clock timing event without substitution of said value to said variable at said clock timing event.

17. (Currently Amended) The ~~hardware description verifying~~ method according to claim 16, wherein said (a) detecting comprises:

sequentially reading out ~~sentence~~statements of said source program;

deleting a storage content in a data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event; and

detecting said source program portion when said read out ~~sentence~~statement refers to said variable which is not stored in said data signal list storage section, at said clock timing event.

18. (Currently Amended) The ~~hardware description verifying~~ method according to claim 11, wherein said source program portion is a portion in which a variable of a wiring line is referred to at a clock timing event and then a value is substituted to said variable at said clock timing event.

19. (Currently Amended) The ~~hardware description verifying~~ method according to claim 18, wherein said (a) detecting comprises:

sequentially reading out ~~sentence~~statements of said source program;

deleting a storage content in said data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event; and

detecting said source program portion when said read out ~~sentence~~statement refers to said variable which is not stored in said data signal list storage section, at said clock timing event.

20. (Currently Amended) The ~~hardware description verifying~~ method according to claim 11, wherein said processor detects said source program portion in which a logical operator is used and a right operand of the operator type includes a variable with substitution.

21. (Currently Amended) A computer readable medium incorporating a set of instructions for carrying out a program for ~~a~~ hardware description language based circuit design verification, ~~verifying method~~ the program comprising instructions for:

(a) detecting a portion of a source program whose compiled code is different in logic ~~interpretation between a case of compiling said source program using a compiler and a case of~~ from a behavioral synthesis-produced module, said source program representing ~~for~~ hardware description ~~being described~~ in a programming ~~program~~ language; and

(b) alarming existence of said source program portion-

22. (Currently Amended) The ~~program~~ medium according to claim 21, wherein said source program portion is a portion in which a variable of a register type is referred to at a clock timing event after substitution to said variable at said clock timing event.

23. (Currently Amended) The medium~~program~~ according to claim 22, wherein said (a) detecting comprises:

sequentially reading out ~~sentences~~statements of said source program;

deleting a storage content in a data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event; and

detecting said read out ~~sentence~~statement as said source program portion when said read out ~~sentence~~statement refers to said variable which is stored in said data signal list storage section.

24. (Currently Amended) The medium~~program~~ according to claim 21, wherein said source program portion is a portion in which substitution to a variable of a non-overwrite type is carried out twice or more at a clock timing event.

25. (Currently Amended) The medium~~program~~ according to claim 24, wherein said (a) detecting comprises:

sequentially reading out ~~sentences~~statements of said source program;

deleting a storage content in a data signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event and said variable is not stored in said data signal list storage section; and

detecting said source program portion when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event and said variable is stored in said data signal list storage section.

26. (Currently Amended) The medium~~program~~ according to claim 21, wherein said source program portion is a portion in which a variable for a non-register type is referred to at a clock timing event without substitution of said value to said variable at said clock timing event.

27. (Currently Amended) The medium ~~program~~ according to claim 26 wherein said (a) detecting comprises:

sequentially reading out ~~sentences~~statements of said source program;

deleting a storage content in a signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said data signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event; and

detecting said source program portion when said read out ~~sentence~~statement refers to said variable which is not stored in said signal list storage section, at said clock timing event.

28. (Currently Amended) The medium~~program~~ according to claim 21, wherein said source program portion is a portion in which a variable of a wiring line is referred to at a clock timing event and then a value is substituted to said variable at said clock timing event.

29. (Currently Amended) The medium~~program~~ according to claim 28. wherein said (a) detecting comprises:

sequentially reading out ~~sentences~~statements of said source program;

deleting a storage content in said signal list storage section when the read out ~~sentence~~statement indicates a clock boundary;

storing said variable in said signal list storage section when said read out ~~sentence~~statement indicates the substitution to said variable at said clock timing event; and

detecting said source program portion when said read out ~~sentence~~statement refers to said variable which is not stored in said data signal list storage section, at said clock timing event.

30. (Currently Amended) The medium~~program~~ according to claim 21, wherein said processor detects said source program portion in which a logical operator is used and a right operand of the operator type includes a variable with substitution.